

**Amendments to the Claims:**

This listing of claims replaces all prior versions, and listings, of claims in this application.

**Listing of Claims:**

1. 1-22. (Canceled)

23. (Presently Amended) A detection system for detecting intrusive behavior in a session on a computer during an application monitoring phase, said session comprising a plurality of applications invoked on said computer, and said computer having a computer operating system, said detection system comprising:

(a) a plurality of trained neural networks, wherein each trained neural network ~~is~~ has previously been trained during a training phase to identify a pre-determined behavior pattern for a corresponding one of the plurality of applications, and wherein each trained neural network is selected for use in the application monitoring phase based upon performance during a testing phase;

(b) a plurality of application profiles, wherein each application profile comprises a plurality of application data for a corresponding one of the plurality of applications, wherein said application data is collected during the session;

(c) a temporal locality identifier, wherein when one of the plurality of application profiles is sequentially input to a corresponding one of the plurality of trained neural networks the trained neural network outputs a behavior indicator for each of the plurality of data strings in the application profile, and wherein if the behavior indicator meets a pre-determined criteria, a counter is incremented, and wherein if the counter has a high rate of increase the temporal

locality identifier labels the application behavior intrusive, and wherein if a predetermined percentage of application behaviors are intrusive the session behavior is labeled intrusive.

24. (Original) The detection system of claim 23, wherein the pre-determined behavior pattern comprises a non-intrusive behavior.

25. (Previously Presented) The detection system of claim 23, the application data comprises a distance between a sequential mapping of system calls made by a corresponding one of the plurality of applications and a pre-defined string of system calls.

26. (Previously Presented) The detection system of claim 23, wherein the application data comprises a distance between a sequential mapping of object requests made by a corresponding one of the plurality of applications and a pre-defined string of object requests.

27. (Original) The detection system of claim 23, wherein the plurality of application profiles is created by a data pre-processor application.

28. (Original) The detection system of claim 27, wherein the data pre-processor receives input from an auditing system integral to the computer operating system.

29. (Original) The detection system of claim 27, wherein the data pre-processor creates the plurality of second application profiles in real-time.

30. (Original) The detection system of claim 27, wherein the plurality of trained neural networks receive input from the plurality of application profiles in real-time.

31. (Canceled)

32. (Canceled)

33. (Presently Amended) The detection system of claim 23, wherein the plurality of trained neural networks comprises a plurality of backpropagation neural networks.

34. (Presently Amended) The detection system of claim 33, wherein each backpropagation neural network in the plurality of backpropagation neural networks comprises an input layer, a hidden layer and an output layer.

35. (Presently Amended) The detection system of claim 34, wherein a number of nodes in the hidden layer is determined by testing a plurality of cases for each backpropagation neural network in the plurality of backpropagation neural networks and selecting the ~~ease~~ wherein the corresponding backpropagation neural network ~~has~~ having a highest accuracy rate

during the testing phase for use in application monitoring.

36. (Presently Amended) The detection system of claim 23, wherein the plurality of trained neural networks comprises a plurality of recurrent neural networks.

37. (Presently Amended) A method for detecting intrusive behavior in a session on a computer during an application monitoring phase, said session comprising a plurality of applications invoked on said computer, and said computer having a computer operating system, said method comprising the steps of:

(a) training a plurality of neural networks during a training phase, wherein each neural network is trained to identify a pre-determined behavior pattern for a corresponding one of the plurality of applications;

(b) creating a plurality of application profiles, wherein each application profile comprises a plurality of application data for a corresponding one of the plurality of applications, wherein said application data is collected during the session;

(c) performing a temporal locality identifying algorithm, wherein when one of the plurality of application profiles is sequentially input to a corresponding one of the plurality of trained neural networks the trained neural network outputs a behavior indicator for each of the plurality of data strings in the application profile, and wherein if the behavior indicator meets a pre-determined criteria, a counter is incremented, and wherein if the counter has a high rate of increase the temporal locality identifier labels the application behavior intrusive, and wherein if a

predetermined percentage of application behaviors are intrusive the session behavior is labeled intrusive.

38. (Original) The method of claim 37, wherein the pre-determined behavior pattern comprises a non-intrusive behavior.

39. (Previously Presented) The method of claim 37, wherein the application data comprises a distance between a sequential mapping of system calls made by a corresponding one of the plurality of applications and a pre-defined string of system calls.

40. (Previously Presented) The method of claim 37, wherein the application data comprises a distance between a sequential mapping of object requests made by a corresponding one of the plurality of applications and a pre-defined string of object requests.

41. (Original) The method of claim 37, wherein the plurality of application profiles is created by a data pre-processor application.

42. (Original) The method of claim 41, wherein the data pre-processor receives input from an auditing system integral to the computer operating system.

43. (Original) The method of claim 41, wherein the data pre-processor creates the plurality of second application profiles in real-time.

44. (Original) The method of claim 41, wherein the plurality of trained neural networks receive input from the plurality of application profiles in real-time.

45. (Canceled)

46. (Canceled)

47. (Presently Amended) The method of claim 37, wherein the plurality of trained neural networks comprises a plurality of backpropagation neural networks.

48. (Presently Amended) The method of claim 37, wherein each backpropagation neural network in the plurality of backpropagation neural networks comprises an input layer, a hidden layer and an output layer.

49. (Presently Amended) The method of claim 48, wherein a number of nodes in the hidden layer is determined by testing a plurality of cases for each backpropagation neural network in the plurality of backpropagation neural networks and selecting the case wherein the

Serial No.:  
Art Unit:

Attorney's Docket No.: CIG-103  
Page 8

corresponding neural network has a highest accuracy rate.

50. (Presently Amended) The method of claim 37, wherein the plurality of trained neural networks comprises a plurality of recurrent neural networks.